

**PROGRAMACIÓN DECLARATIVA: LÓGICA Y RESTRICCIONES**

**Grado en Ingeniería Informática / Grado en Matemáticas e Informática**

**7 de Junio de 2018**

**Examen de Evaluación Continua**

**Apellidos:**

**Nombre:**

**Nº de matrícula:**

**DURACIÓN:** 60 minutos

**INSTRUCCIONES:** Todas las preguntas se deben contestar exclusivamente en las hojas de examen, en los espacios habilitados para tal efecto.

**Pregunta 1.** (2,5 puntos)

**1.a.** Escribir en **programación lógica pura** el predicado **trenzar(L1,L2,LT)** (**trenzar/3**) que se verifica si LT es la lista que se obtiene combinando los elementos de L1 y L2 de la siguiente manera: un elemento de L1 y dos de L2, sucesivamente hasta utilizar todos los elementos de ambas listas. Se supone que L2 tiene el doble de elementos que L1.

**SOLUCIÓN:**

trenzar([], [], []).

trenzar([X|Xs],[Y,Z|Ys],[X,Y,Z|LTs]) :-  
trenzar(Xs,Ys,LTs).

**1.b.** Escribir la **consulta** necesaria para saber que dos listas se usan para obtener la trenza representada por la lista [1,a,b,2,c,d,3,e,f].

?- trenzar(A,B,[1,a,b,2,c,d,3,e,f]).

A = [1,2,3],

B = [a,b,c,d,e,f] ? ;

no

**1.c.** Suponiendo el siguiente código:

sinNombre([ficha(A,B)|Xs],ficha(A,B),Xs).

sinNombre([Y|Xs],ficha(A,B),[Y|Zs]) :-

sinNombre(Xs, ficha(A,B),Zs).

Proporcionar **todas las respuestas (una por línea)** a las siguientes consultas:

?- sinNombre([],ficha(a,b),L).

no

?- sinNombre([ficha(a,b)],ficha(a,b),L).

L = [] ? ;

no

?- sinNombre([ficha(a,b),s(0),ficha(a,b),s(s(0)),ficha(a,e)],ficha(a,b),L).

L = [s(0),ficha(a,b),s(s(0)),ficha(a,e)] ? ;

L = [ficha(a,b),s(0),s(s(0)),ficha(a,e)] ? ;

no

## Pregunta 2. (3 puntos)

2.a. Dado el siguiente programa lógico:

```
obtenerDimensiones(Elto,Dim) :-  
    compound(Elto),  
    Elto =.. [_Tipo|Dim].
```

Proporcionar **todas las respuestas (una por línea)** a las siguientes consultas:

?- obtenerDimensiones(3,X).

no

?- obtenerDimensiones(rectangulo(3,5),X).

X = [3,5] ? ;

no

2.b. Dado el siguiente programa lógico:

```
estaEnAlineacion(X) :- \+lesionado(X), entrenadorConfiaEn(X).  
entrenadorConfiaEn(X) :- entrenaBien(X).  
entrenadorConfiaEn(X) :- buenaActitud(X).  
lesionado(carvajal).  
lesionado(benzema).  
entrenaBien(ramos).  
entrenaBien(marcelo).  
buenaActitud(bale).  
buenaActitud(benzema).
```

- Proporcionar **todas las respuestas (una por línea)** a la pregunta “estaEnAlineacion(X)”.

no

- Modificar el programa anterior** para que a la pregunta "estaEnAlineacion(X)" el intérprete suministre únicamente la respuesta "X = ramos". Solamente está permitido reordenar clausulas y/u objetivos y el uso de predicados de control de flujo.

### SOLUCIÓN 1:

```
estaEnAlineacion(X) :- entrenadorConfiaEn(X), \+lesionado(X), !.
```

**PROGRAMACIÓN DECLARATIVA: LÓGICA Y RESTRICCIONES**

**Grado en Ingeniería Informática / Grado en Matemáticas e Informática**

**7 de Junio de 2018**

**Examen de Evaluación Continua**

**Apellidos:**

**Nombre:**

**Nº de matrícula:**

**DURACIÓN:** 60 minutos

**INSTRUCCIONES:** Todas las preguntas se deben contestar exclusivamente en las hojas de examen, en los espacios habilitados para tal efecto.

**SOLUCIÓN 2:**

**estaEnAlineacion(X) :- entrenadorConfiaEn(X), !, \+lesionado(X).**

**Pregunta 3.** Suponiendo el siguiente programa lógico: (2,5 puntos)

```
redCarreteras([carretera(a,c,autopista,10),
               carretera(b,a,autovia,5),
               carretera(b,c,cNacional,35),
               carretera(d,b,cComarcal,55),
               carretera(d,e,cLocal,45),
               carretera(c,e,autovia,12),
               carretera(c,d,autovia,24),
               carretera(e,b,autovia,15),
               carretera(e,d,cLocal,12)]).
```

**vias(I,T):-**

```
arg(1,I,Tipos),
setof(X,(member(X,Tipos)),T).
```

**3.a.** Proporcionar **todas las respuestas (una por línea)** a la siguiente pregunta:

"findall((A,B),(redCarreteras(R),member(C,R),C=..[carretera,A,B,cLocal,\_L]),CConectadas)"

**CConectadas = [(d,e),(e,f)] ? ;**

**no**

**3.b.** Proporcionar **todas las respuestas (una por línea)** a la siguiente consulta:

?- vias(itinerario([autovia,autovia,cNacional,autovia,cLocal,cLocal],\_X),T).

**T = [autovia,cLocal,cNacional] ? ;**

**no**

**PROGRAMACIÓN DECLARATIVA: LÓGICA Y RESTRICCIONES**

**Grado en Ingeniería Informática / Grado en Matemáticas e Informática**

**7 de Junio de 2018**

**Examen de Evaluación Continua**

**Apellidos:**

**Nombre:**

**Nº de matrícula:**

**DURACIÓN:** 60 minutos

**INSTRUCCIONES:** Todas las preguntas se deben contestar exclusivamente en las hojas de examen, en los espacios habilitados para tal efecto.

**Pregunta 4.** Proporcionar **todas las respuestas (una por línea)** a las siguientes consultas: (2 puntos)

?- X is 5, Y is X+5.

X = 5, Y = 10

?- Y is X+3, X is 5.

Error (porque X está libre)

?- 1+1 is 2.

no (la parte izquierda no unifica con 2)

?- X = s(0).

no

?- X =:= X.

{ERROR: arithmetic:=:=/2 - instantiation\_error}

?- functor(3+4,X,Y).

X = +, Y = 2

?- arg(5, [a,b,c,d,e], E).

no (porque no existe el argumento 5)

Suponiendo el siguiente programa lógico:

`division(X, Y, Z) :- Z is X // Y.`

`division2(X, Y, Z) :- Z =. X // Y`

Proporcionar **todas las respuestas (una por línea)** a las siguientes consultas:

?- X is 10+2, division2(X,3, Z).

`X = 12, Z = 4`

?- division2(X, 4, 1).

`X = 4`

?- division2(X, Y, 12).

`X = 12 * Y`